CS206 Tutorial No. #4

Date: Feb 17, 2006

1. Consider the formula
   $\forall(x)[P(x) \Leftrightarrow \forall(y)[Q(x,\ y) \Leftrightarrow (eq(y,\ f_1()) \vee eq(y,\ x))\ ]\ ]$
   Evaluate the formula in the following models: $Q(x,\ y)\ \equiv\ y$ *divides* x, $eq(x,\ y) \equiv\ x =\ y$ and $f_1()$ is a *zero-ary* function which evaluates to the constant value 1 and

   (a) Domain=$\{1, 2, 4, 8\}$, and $P(x) \equiv$ *x is a composite number*

   (b) Domain=$\{1, 2, 4, 8\}$, and $P(x) \equiv$ *x is a prime number*

   (c) Domain=$\{1, 2, 3,\ \ldots\}$, and $P(x) \equiv$ *x is a prime number*

   (d) Domain=$\{1, 2, 3,\ \ldots\}$, and $P(x) \equiv$ *x is a composite number*

2. In this question we wish to reason about lists using predicate logic. Note that a list is an ordered sequence *(not a set)* of elements. You are allowed to use the following function and predicate symbols in your formulae. The intent of each these functions and predicates is as indicated by their names.

   - Predicate symbols: $is\_empty\_list(x)$, $is\_list\_with\_one\_element(x)$, and $equal(x)$.
   - Function symbols: $reverse\_list(x)$ and $append\_list\_to\_list(x, y)$.

   Using only the above predicate and function symbols and predicate logic operators, express the following English language statements as predicate logic formulae. You can assume that the universe, S, of a model contains elements that are only lists.

   (a) Every list can be obtained by appending a list to another list.

   (b) For any list x, whenever it is possible to express it as the append of list y and list z, it can be reversed by appending the reverse of list z to reverse of list y.

   (c) A list is empty iff it keeps every list unchanged after it is appended to the list.

   (d) There are lists, not all of whose elements are identical.